OXFORD

Briefings in Bioinformatics, 2025, **26(3)**, bbaf278
https://doi.org/10.1093/bib/bbaf278
Problem Solving Protocol

GC-balanced polar codes correcting insertions, deletions and substitutions for DNA storage

Rui Zhang[1] and Huaming Wu[2,*]

[1]Chern Institute of Mathematics, Nankai University, 94 Weijin Road, 300071 Tianjin, China
[2]Center for Applied Mathematics, Tianjin University, 92 Weijin Road, 300072 Tianjin, China

*Corresponding author. E-mail: whming@tju.edu.cn

Abstract

In order to address the insertion, deletion, and substitution (IDS) errors inherent in deoxyribonucleic acid (DNA) storage channels during DNA synthesis and sequencing, we propose a novel GC-balanced polar code scheme tailored to rectify these errors by incorporating the unique characteristics of the DNA storage channel into the polar code design. The innovation lies in modeling errors as a drift vector, reflecting deviations from the desired DNA sequence, aiming to improve the reliability of DNA-based data storage. In this paper, we developed a GC-balanced polar code scheme named DNA-BP Code, which stands for balanced polar code for DNA storage, that effectively rectifies IDS errors in DNA storage. The computational complexity of the proposed encoding and decoding algorithms is $\mathcal{O}(N \log N)$ with respect to the code length N. Simulation results show the bit error rate and block error rate as functions of the code length and IDS probability, demonstrating the efficacy of our approach in enhancing the accuracy of DNA storage systems.

Keywords: DNA storage; polar code; GC-balance; IDS errors

Introduction

Currently, with the rapid development of information technology and the widespread use of social networking, the demand for global data storage has exceeded its current capacity. Deoxyribonucleic acid (DNA) molecules, renowned for carrying natural genetic information, emerge as a stable, resource-efficient, and sustainable solution for data storage [1–3]. However, the methods for synthesizing and sequencing DNA sequences are still imperfect and struggle to maintain pace with the necessity for accurate DNA storage [4, 5]. Common errors encountered during DNA synthesis and sequencing include insertions, deletions, and substitutions (IDS). In addition to addressing IDS errors, another critical design consideration in DNA coding is GC-content balancing. Recent studies have shown that maintaining a balanced ratio of G-C (guanine-cytosine) and A-T (adenine-thymine) nucleotides can significantly reduce synthesis and sequencing errors in DNA storage systems [6–9]. Specifically, DNA sequences with minimal GC imbalance demonstrate improved physical stability during both storage and retrieval processes. While completely balanced GC content (zero imbalance) would be ideal, research indicates that achieving a small, bounded imbalance of order $\mathcal{O}(N)$ is sufficient for practical error reduction in DNA data storage applications.

Consequently, the primary obstacle in implementing DNA storage lies in designing effective error-correcting codes capable of rectifying these errors with high reliability. Numerous sophisticated error correction methods have been proposed to address this challenge [10–14].

Among various error-correcting codes, the polar code with successive cancelation (SC) decoding, introduced by Arikan [15], has garnered significant attention. Şaşoğlu *et al.* [16] demonstrated that this code is the only method capable of achieving the symmetric capacity of any binary-input discrete memoryless channel with a computational complexity of $N \log N$, where N denotes the code length. This remarkable property positions polar codes as a promising approach for addressing errors in DNA storage. However, traditional polar codes are designed for binary-input discrete memoryless channels (B-DMC), whereas the channels encountered in DNA storage exhibit memory. Therefore, it is necessary to enhance traditional polar codes to adapt them for DNA storage applications. Thomas *et al.* [17] proposed a polar encoding scheme tailored for erasure and deletion channels, primarily addressing single deletion errors. Additionally, Tian *et al.* [18] introduced SC decoding of polar codes for channels with d deletions, demonstrating its capability to achieve symmetric capacity.

In traditional IDS channels, assuming that insertions and deletions happen probabilistically, rather than in isolation, is typical. This means that a received word may contain multiple insertion and deletion errors simultaneously, making it challenging to uniquely determine the number of insertions and deletions based solely on the length of the received word. Consequently, extending polar coding for d-deletions to accommodate multiple insertion/deletion error correction codes is not straightforward.

The main contributions of this letter are three-fold:

- We introduce a novel polar code design, termed the DNA-BP code, an acronym for "balanced polar code for DNA storage."

Received: January 13, 2025. **Revised:** April 8, 2025. **Accepted:** May 22, 2025

This scheme is specifically engineered to address and correct the IDS errors that are prevalent in DNA storage systems.
- We have enhanced the traditional SC decoding method specifically for DNA storage, resulting in a new decoding approach that is faster and more accurate.
- The novel polar code is GC-balanced to deal with GC-content constraints inherent in DNA storage systems.

Methods

In this work, we adopt specific notations as follows: the code length is represented by $N = 2^n$. We define finite subsets of integers as $\mathbb{Z}_M = \{0, 1, \ldots, M-1\}$, $\mathbb{B} = \mathbb{Z}_2 = \{0, 1\}$, $\mathbb{D} = \mathbb{Z}_4 = \{A, T, C, G\}$. Bold letters are utilized to denote sequences, while plain letters signify symbols within those sequences. For instance, $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1})$. In the case of a binary vector $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1}) \in \mathbb{B}^N$ of length N, the sub-vector $\mathbf{x}_i^j$ is defined as:

$$\mathbf{x}_i^j = \begin{cases} (x_i, x_{i+1}, \ldots, x_j), & 0 \le i \le j \le N-1, \\ \epsilon, & \text{otherwise}, \end{cases} \tag{1}$$

where ϵ is the vector of length zero.

Since the IDS errors occupy the majority of DNA storage errors during DNA synthesis and sequencing operations, we denote the probabilities of these errors as p_i, p_d, and p_s, respectively. To effectively model DNA storage operations, we consider DNA alphabet sequences transmitted through a specific channel that encompasses IDS errors. We refer to this channel as the IDS channel.

Let $\mathbf{x}$ denote a binary word of length N. We utilize $\mu(\mathbf{x})$ to represent its imbalance. In logarithmic expressions with a base of 2, we express $\log N$ as shorthand for $\log_2 N$ throughout this paper. For DNA storage, we represent the alphabet by $\mathbb{Z}_4 = \{A, T, C, G\}$. Let $\sigma = (\sigma_0, \sigma_1, \cdots, \sigma_{N-1}) \in \mathbb{Z}_4^N$ denote the input vector, and $\tau = (\tau_0, \tau_1, \cdots, \tau_{N'-1}) \in \mathbb{Z}_4^{N'}$ denote the channel output. We also establish a one-to-one bijection Φ between $\mathbb{Z}_4$ and two-bit sequences as follows:

$$A \leftrightarrow 00, \quad T \leftrightarrow 01, \quad C \leftrightarrow 10, \quad G \leftrightarrow 11. \tag{2}$$

Thus, given any DNA nucleotides sequence $\sigma \in \mathbb{Z}_4^N$, we will have a corresponding binary sequence $\mathbf{x} \in \{0,1\}^{2N} = \mathbb{Z}_2^{2N}$, and we write $\mathbf{x} = \Phi(\sigma)$, where $\mathbf{x} = \mathbf{x}_0^{2N-1} = (x_0, x_1, \cdots, x_{2N-1}) \in \mathbb{Z}_2^{2N}$.

Given two sequences $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1})$ and $\mathbf{y} = (y_0, y_1, \ldots, y_{N-1})$, we denote the concatenation of the two sequences as $\mathbf{xy}$. In the special case where $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_2$, we use $\mathbf{x}||\mathbf{y}$ to represent their interleaved sequence $x_0 y_0 x_1 y_1 \ldots x_{N-1} y_{N-1}$. The XOR of binary vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as:

$$\mathbf{x} \oplus \mathbf{y} = (x_0 \oplus y_0, x_1 \oplus y_1, \ldots, x_{N-1} \oplus y_{N-1}). \tag{3}$$

Definition 1. For any DNA nucleotide sequence $\boldsymbol{\sigma} \in \mathbb{Z}_4^N$, a corresponding binary sequence $\mathbf{x} \in \{0,1\}^{2N} = \mathbb{Z}_2^{2N}$ can be obtained, represented as $\mathbf{x} = \Phi(\sigma)$, where $\mathbf{x} = \mathbf{x}_0^{2N-1} = (x_0, x_1, \ldots, x_{2N-1}) \in \mathbb{Z}_2^{2N}$. Define $E_{\boldsymbol{\sigma}} = x_0 x_2 \cdots x_{2N-2}$, $O_{\boldsymbol{\sigma}} = x_1 x_3 \cdots x_{2N-1}$; thus, $\Phi(\boldsymbol{\sigma}) = E_{\boldsymbol{\sigma}} || O_{\boldsymbol{\sigma}}$. The sequences $E_{\boldsymbol{\sigma}}$ and $O_{\boldsymbol{\sigma}}$ are referred to as the even-indexed sequence and odd-indexed sequence of $\boldsymbol{\sigma}$, respectively.

Example 1. Given $\sigma = ACATAG$, then $\mathbf{x} = \Phi(\sigma) = 001000010011$, then even index sequence and odd index sequence of σ are $E_\sigma = 000101$ and $O_\sigma = 010001$.

Definition 2. For a binary vector $\mathbf{v} = (v_0, v_1, \ldots, v_{2N-1}) \in \mathbb{B}^{2N}$ of even length $2N$, the definitions of $E(\mathbf{v})$ and $O(\mathbf{v})$ are as follows:

$$E(\mathbf{v}) = (v_0, v_2, \ldots, v_{2i}, \ldots, v_{2N-2}), \tag{4}$$

$$O(\mathbf{v}) = (v_1, v_3, \ldots, v_{2i+1}, \ldots, v_{2N-1}), \tag{5}$$

where i is any non-negative integer.

The probability of a random variable X taking the value x is denoted as $p(x) \triangleq \Pr(X = x)$. Conditional and joint probabilities are denoted as $p(x|y) \triangleq \Pr(X = x|Y = y)$ and $p(x, y) \triangleq \Pr(X = x, Y = y)$.

Let $\mathbf{x}$ and $\mathbf{y}$ denote the transmitted and received sequences, respectively, where $\mathbf{x} = x_0 x_1 \ldots x_{N-1} \in \mathbb{B}^N$ and $\mathbf{y} = y_0 y_1 \ldots y_{N'-1} \in \mathbb{B}^{N'}$. According to [19], insertion and deletion errors between $\mathbf{x}$ and $\mathbf{y}$ are expressed by the drift vector:

$$\mathbf{d} = (d_0, d_1, \ldots, d_n - 1, d_N) \in \mathcal{D}^{N+1}, \tag{6}$$

where D represents the maximum absolute value of drift between $\mathbf{x}$ and $\mathbf{y}$, and $\mathcal{D} = \{-D, \cdots, -1, 0, 1, \cdots, D\}$ denotes the set of drift values.

The drift vector is determined by the Markov process in our hypothesis, with the following state transition probabilities:

$$p(d_{i+1}|d_i) = \begin{cases} p_i, & d_{i+1} = d_i + 1 \ \& \ d_i \neq D, \\ p_d, & d_{i+1} = d_i - 1 \ \& \ d_i \neq -D, \\ 1 - p_i - p_d, & d_{i+1} = d_i \ \& \ -D < d_i < D, \\ 1 - p_i, & d_{i+1} = d_i \ \& \ d_i = -D, \\ 1 - p_d, & d_{i+1} = d_i \ \& \ d_i = D, \\ 0, & \text{otherwise}, \end{cases}$$

where the initial drift value is $d_0 = 0$.

Let $S(i; d_i, d_{i+1}) = \{i' \mid i + d_i \le i' < (i+1) + d_{i+1}\} \subset \mathbb{Z}_{N'}$, then, we have:

$$|S(i; d_i, d_{i+1})| = \begin{cases} 2, & d_{i+1} = d_i + 1 (\textit{insertion}), \\ 1, & d_{i+1} = d_i, \\ 0, & d_{i+1} = d_i - 1 (\textit{deletion}). \end{cases} \tag{7}$$

We present our DNA-BP code scheme for DNA storage. First, we provide an overview of the encoding and decoding workflow in Fig. 1. Then, we detail the GC-balanced polar encoder in Fig. 2 and the modified SC decoder in Fig. 3.

Given a message sequence $\mathbf{m} \in \mathbb{Z}_2^{2N}$, and considering the bijection between $\mathbb{Z}_2^{2N}$ and $\mathbb{Z}_4^N$, we define $\sigma \in \mathbb{Z}_4$ as the DNA nucleotide representation sequence of the encoded sequence $\mathbf{x}$. Then, E_σ and O_σ represent the even index sequence and odd index sequence of σ, respectively. Treating E_σ and O_σ as two distinct sequences, we encode and decode them separately. This approach allows us to analyze the encoding and decoding

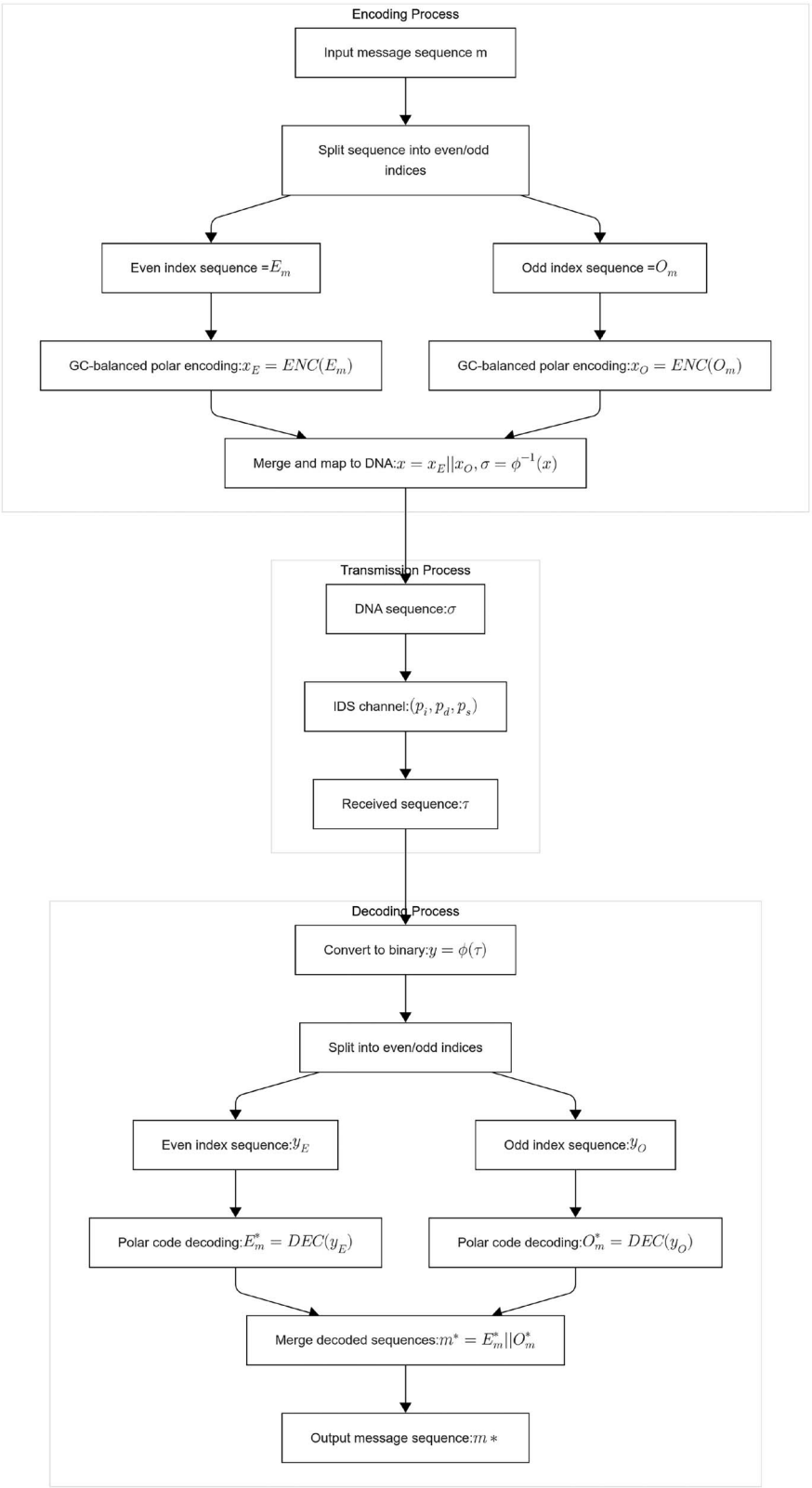


Figure 1. The workflow of DNA-BP code encoding and decoding.

problem in the quaternary system within the binary system framework.

Let the length of the polar code under consideration be $N = 2^n$. Additionally, let $\mathcal{A}$ and $\mathcal{A}^C$ denote sets of information and frozen bits, respectively, where $\mathcal{A} \cap \mathcal{A}^C = \emptyset$ and $\mathcal{A} \cup \mathcal{A}^C = \mathbb{Z}_N$.

Encoding

Different from the traditional encoding scheme of Arikan, we adopt the encoding scheme from [20], where it has been proved that we can significantly reduce the imbalance of all codewords to the smallest imbalance by sacrificing only $\log N$ information bits

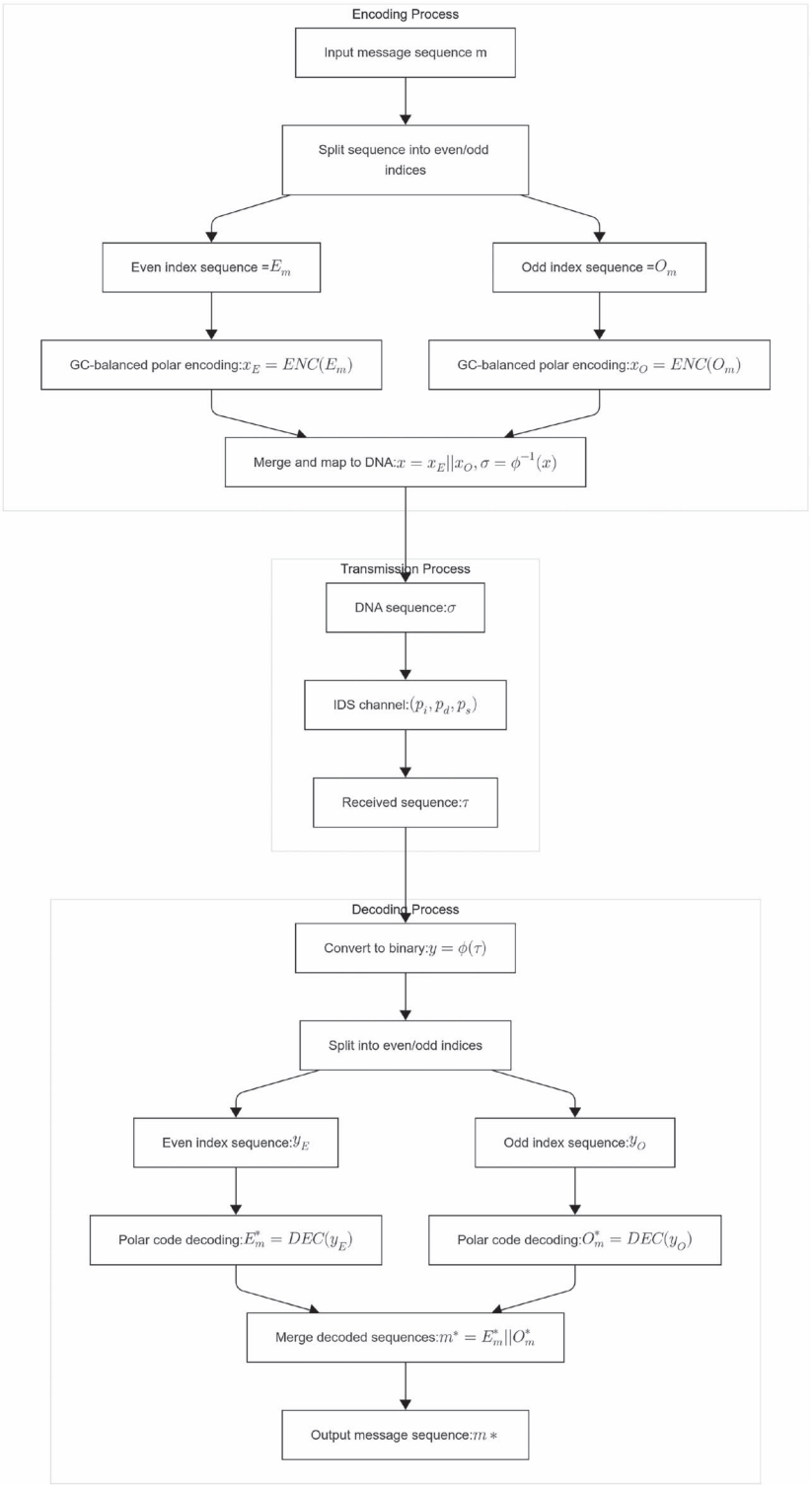


Figure 2. The workflow of GC-balanced polar encoder.

in $\mathcal{O}(N \log N)$ time. As mentioned in the introduction, for error reduction in DNA data storage, it suffices to have DNA strings with minimal GC imbalance [6–8]. Therefore, rather than aiming for an imbalance of 0, our focus is on finding a code where the imbalance is $\mathcal{O}(N)$. Let the information word be $\mathbf{m} = \mathbf{u}_0^{N-1}$, which will be encoded as $\mathbf{x}_0^{N-1}$ using the following calculation:

$$x_0^{N-1} = u_0^{N-1} \mathrm{B_N F}^{\otimes n}, \tag{8}$$

where $B_N = R_N \left(I_2 \otimes B_{N/2}\right)$, and R_N is the permutation matrix and its effect is as follows:

$$\begin{aligned}&(u_0, u_1, u_2, u_3, u_4, \cdots, u_{N-1}) \times R_N \\ &= (u_1, u_3, u_5, \cdots, u_{N-1}, u_0, u_2, u_4, \cdots, u_{N-2}.)\end{aligned} \tag{9}$$

According to [20], we can pick a subset $\mathcal{B}$ from the information set $\mathcal{A}$ to reduce the imbalance of codewords. Specifically, we can

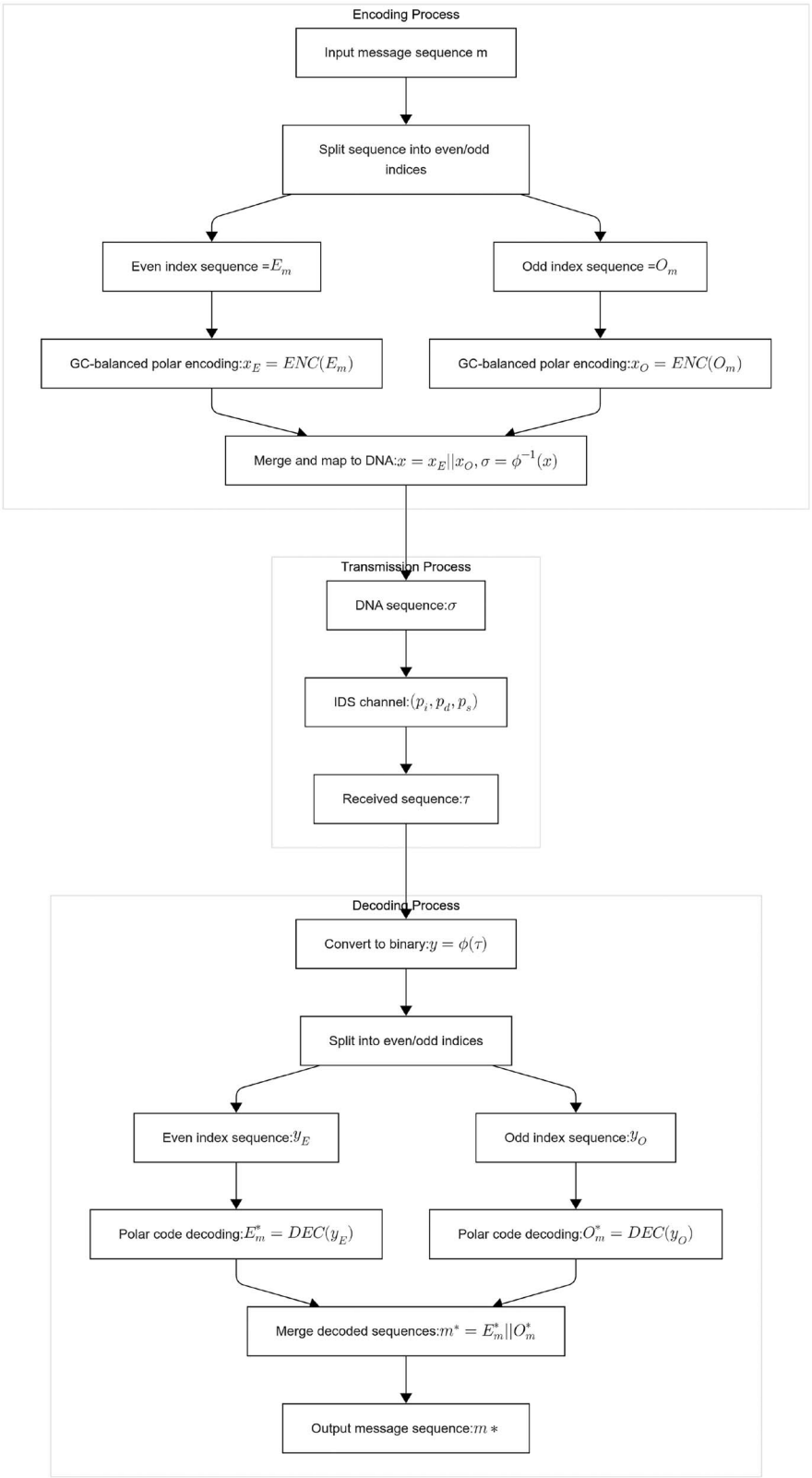


Figure 3. The workflow of modified SC decoder for IDS channels.

reduce the number of information bits to $k' = N - |\mathcal{A}^C| - |\mathcal{B}|$ so that a k'-bit message $\mathbf{m}'$ is encoded to a polar codeword $\mathbf{c} = \mathbf{x}_0^{N-1}$ with small imbalance $\mu(\mathbf{c})$. We pick a set of balancing indices $\mathcal{B}$ so that $\mathcal{A}^C \cap \mathcal{B} = \emptyset$ and let $\mathbb{B}$ be the linear span of the rows corresponding to $\mathcal{B}$. We transmit k'-bit messages(instead of k-bit, $k = N - |\mathcal{A}^C|$). We first insert $|\mathcal{B}| = k - k'$ zeros to $\mathbf{m}'$ at positions corresponding to $\mathcal{B}$ and compute the corresponding encoding $\mathbf{c}'$. Next, we find a balancing vector $\mathbf{b} \in \mathcal{B}$ so that the corresponding imbalance $\mu(\mathbf{c}' + \mathbf{b})$ is minimized. Then, we transmit the word $\mathbf{c} \triangleq \mathbf{c}' + \mathbf{b}$. For more details, see Algorithm 2. Let the GC balanced polar encoder

Algorithm 1: Framework of polar code for DNA storage.

Input: Message DNA sequence: $\mathbf{m} = m_0 m_1 \cdots m_{2N}$; Error rate of IDS channel p_i, p_d, p_s

Output: prediction sequence: $\mathbf{m}^*$.

1: Divide the sequence $\mathbf{m}$ into two parts according to the parity of index.
2: $E_m = m_0 m_2 \cdots m_{2N-2}$, $O_m = m_1 m_3 \cdots m_{2N-1}$
3: $\mathbf{m} = E_m || O_m$
4: Encode E_m and O_m separately in polar code.
5: $\mathbf{x}_E = ENC(E_m) = x_0 x_2 \cdots x_{2N'-1}$
6: $\mathbf{x}_O = ENC(O_m) = x_1 x_3 \cdots x_{2N'}$
7: Merge and map to DNA sequences.
8: $\mathbf{x} = \mathbf{x}_E || \mathbf{x}_O$
9: $\sigma = \Phi^{-1}(\mathbf{x})$
10: DNA sequence σ goes throught the IDS channel with error rates p_i, p_d and p_s, and outputs sequence τ and divide the corresponding binary sequence $\mathbf{y}$ into two parts according to the parity of index.
11: $\mathbf{y} = \Phi(\tau) = \mathbf{y}_E || \mathbf{y}_O$
12: Decode $\mathbf{y}_E$ and $\mathbf{y}_O$ with polar code decoder.
13: $E_m^* = DEC(\mathbf{y}_E)$, $O_m^* = DEC(\mathbf{y}_O)$
14: Merge.
15: $\mathbf{m}^* = E_m^* || O_m^*$
16: Output $\mathbf{m}^*$

be denoted as ENC, then we have $\mathbf{c} = ENC(m')$, and the imbalance $\mu(\mathbf{c})$ is small.

Algorithm 2: Framework of GC balanced polar encoder

Input: Message sequence $\mathbf{m}' \in \mathbb{Z}_2^{k'}$
Output: Encoded codeword $\mathbf{c} \in \mathbb{Z}_2^N$
initialization;
if $\mathcal{A} \neq \emptyset$ **then**
  pick $\mathcal{B}$, $\mathcal{A} \cap \mathcal{B} = \emptyset$;
  $k = k' + |\mathcal{B}|$;
  insert $k - k'$ zeros to $\mathbf{m}'$ at positions corresponding to $\mathcal{B}$, $\mathbf{m}' \in \mathbb{Z}_2^k$;
  $\mathbf{c}' \leftarrow \mathbf{m}'$
else
  fix a set of frozen indices $\mathcal{A}$.
while $|\mathbf{c}'| = k$ **do**
  find $\mathbf{b} \in \mathbb{B}$, so that $\mu(\mathbf{c}' + \mathbf{b})$ is minimized, where $\mathbb{B}$ is the linear span of rows corresponding to $\mathcal{B}$
  $\mathbf{c} = \mathbf{c}' + \mathbf{b}$

Decoding

To decode a noisy word $\tilde{\mathbf{c}}$, we can simply apply a slightly modified polar-decoding algorithm and find the k-bit vector $\mathbf{m} = DEC(\tilde{\mathbf{c}})$. The desired message $\hat{\mathbf{m}}'$ is then the k'-prefix of $\mathbf{m}$. That is to say, if $\mathbf{m}$ is successfully decoded under the polar coding scheme, we also successfully recover our message $\mathbf{m}'$. We use a slightly modified SCL decoding method to deal with the codewords, which has been given the frozen bit set $\mathcal{A}^C \subset \mathbb{Z}$ and the values of the frozen bits, and details of the IDS channel parameters, p_i, p_d, p_s, and D. The decoder inputs received word $\mathbf{y}_0^{N'-1} \in \mathbb{B}^{N'}$, and outputs decoded word $\tilde{\mathbf{u}}_0^{N-1} = (\tilde{u}_0, \tilde{u}_1, \ldots, \tilde{u}_{N-1}) \in \mathbb{B}^N$. We modified the traditional SCL decoding method to deal with the Markov drift value d_i as follows.

We use the following formula to estimate the ith information bit u_i:

$$\hat{u}_i = \begin{cases} h_i\left(N', \mathbf{y}_0^{N'-1}, \hat{\mathbf{u}}_0^{i-1}\right), & if\ i \in \mathcal{A} \\ u_i, & if\ i \in \mathcal{A}^c \end{cases} \tag{10}$$

where

$$h_i\left(N', \mathbf{y}_0^{N'-1}, \hat{\mathbf{u}}_0^{i-1}\right) = \begin{cases} 0, & if\ L_N^{(i)}\left(d_N, \mathbf{y}_0^{N'-1}, \hat{\mathbf{u}}_0^{i-1}\right) \geq 0 \\ 1, & if\ L_N^{(i)}\left(d_N, \mathbf{y}_0^{N'-1}, \hat{\mathbf{u}}_0^{i-1}\right) < 0 \end{cases}$$

and

$$L_N^{(i)}\left(d_N, \mathbf{y}_0^{N'-1}, \hat{\mathbf{u}}_0^{i-1}\right) \triangleq \ln\left(\frac{W_N^{(i)}\left(d_N, \mathbf{y}_0^{N'-1}, \hat{\mathbf{u}}_0^{i-1} | u_i = 0\right)}{W_N^{(i)}\left(d_N, \mathbf{y}_0^{N'-1}, \hat{\mathbf{u}}_0^{i-1} | u_i = 1\right)}\right) \tag{11}$$

We can now consider the polar bit IDS channel of level $k = n$ as follows:

$$\begin{aligned} &W_{2^n}^{(i)}\left(d_N, \mathbf{y}_0^{N'-1}, \mathbf{u}_0^{i-1} | d_0 = 0, u_i\right) \\ &= p(d_N, \mathbf{y}_0^{N'-1}, \mathbf{u}_0^{i-1} | d_0 = 0, u_i) \end{aligned} \tag{12}$$

Consistent with the traditional polar code decoding method, we need to find a recursion method so that the probability of level k can be calculated using the probability of level $k-1$. The difference is that we need to take d_i into account when calculating the probability.

Recuisions for level $k \in \{1, 2, \ldots, n\}$

We first list notations that are used in our recursion:

$$\begin{aligned} &a = 2^k m, b = 2^k(m+1), c = (a+b)/2, \\ &\tilde{\mathbf{v}} = (\tilde{v}_0, \tilde{v}_1, \ldots, \tilde{v}_{2^{k-1}-1}) = \mathbf{u}(k-1)_a^{c-1} \in \mathbb{Z}_2^{2^k-1}, \\ &\tilde{\mathbf{w}} = (\tilde{w}_0, \tilde{w}_1, \ldots, \tilde{w}_{2^{k-1}-1}) = \mathbf{u}(k-1)_c^{b-1} \in \mathbb{Z}_2^{2^k-1}, \\ &\tilde{\mathbf{u}} = \tilde{\mathbf{v}} + \tilde{\mathbf{w}} = (\tilde{v}_0, \tilde{v}_1, \ldots, \tilde{v}_{2^{k-1}-1}) = \mathbf{u}(k-1)_a^{b-1} \in \mathbb{Z}_2^{2^k}, \\ &\mathbf{e} = E(\tilde{\mathbf{u}}_0^{2j-1}) \oplus O(\tilde{\mathbf{u}}_0^{2j-1}) \in \mathbb{Z}_2^j, \\ &\mathbf{f} = O(\tilde{\mathbf{u}}_0^{2j-1}) \in \mathbb{Z}_2^j. \end{aligned}$$

Now, we can calculate the probabilities for bits of even indices as follows:

$$\begin{aligned} &W_{2^k}^{(2j)}\left(d_b, \mathbf{y}_{a+d_a}^{b+d_b-1}, \tilde{\mathbf{u}}_0^{2j-1} \mid d_a, \tilde{u}_{2j}\right) \\ &= \frac{1}{2} \sum_{d_c \in \mathcal{D}} \sum_{\tilde{u}_{2j+1} \in \mathbb{Z}_2} W_{2^{k-1}}^{(j)}\left(d_c, \mathbf{y}_{a+d_a}^{c+d_c-1}, \tilde{\mathbf{v}}_0^{j-1} = \mathbf{e} \mid d_a, \tilde{v}_j = \tilde{u}_{2j} \oplus \right. \\ &\quad \left. \tilde{u}_{2j+1}\right) \times W_{2^{k-1}}^{(j)}\left(d_b, \mathbf{y}_{c+d_c}^{b+d_b-1}, \tilde{\mathbf{w}}_0^{j-1} = \mathbf{f} \mid d_c, \tilde{w}_j = \tilde{u}_{2j+1}\right), \end{aligned} \tag{13}$$

The odd indices are calculated similarly as follows. Due to limited space, we omit the calculation process:

$$\begin{aligned} &W_{2^k}^{(2j+1)}\left(d_b, \mathbf{y}_{a+d_a}^{b+d_b-1}, \tilde{\mathbf{u}}_0^{2j} \mid d_a, \tilde{u}_{2j+1}\right) \\ &= \frac{1}{2} \sum_{d_c \in \mathcal{D}} W_{2^{k-1}}^{(j)}\left(d_c, \mathbf{y}_{a+d_a}^{c+d_c-1}, \tilde{\mathbf{v}}_0^{j-1} = \mathbf{e} \mid d_a, \tilde{v}_j = \tilde{u}_{2j} \oplus \right. \\ &\quad \left. \tilde{u}_{2j+1}\right) \times W_{2^{k-1}}^{(j)}\left(d_b, \mathbf{y}_{c+d_c}^{b+d_b-1}, \tilde{\mathbf{w}}_0^{j-1} = \mathbf{f} \mid d_c, \tilde{w}_j = \tilde{u}_{2j+1}\right). \end{aligned} \tag{14}$$

where $p\left(\tilde{u}_{2j} \mid \tilde{u}_{2j+1}\right) = p\left(\tilde{u}_{2j+1} \mid \tilde{u}_{2j}\right) = \frac{1}{2}$

Calculation for level $k = 0$

For $i \in \{0, 1, \ldots, N-1\}$, the probability is calculated as

$$W_{2^0}^{(i)}\left(d_{i+1}, \mathbf{y}_{i+d_i}^{(i+1)+d_{i+1}-1} \mid d_i, \mathbf{u}(0)_i\right) = p\left(\mathbf{y}_{i+d_i}^{i+d_{i+1}} \mid d_i, d_{i+1}, x_i\right) \cdot p(d_{i+1} \mid d_i), \tag{15}$$

where the second factor of the right-hand side is given already, and the first factor is calculated as:

$$p\left(\mathbf{y}_{i+d_i}^{i+d_{i+1}} \mid d_i, d_{i+1}, x_i\right) = \begin{cases} p_s^{\delta}(1-p_s)^{l-\delta}, & |d_{i+1} - d_i| \le 1, i + d_i \ge 0, i + d_{i+1} < N', \\ 0, & \text{otherwise}, \end{cases} \tag{16}$$

where $l = |S(i; d_i, d_{i+1})|$, and

$$\delta = |i' \in S(i; d_i, d_{i+1}) \mid y_{i'} \neq x_i| = \sum_{i' \in S(i; d_i, d_{i+1})} (x_i \oplus y_{i'}).$$

Determination of frozen bits

Let $I(W_N^{(i)})$ denote the symmetric capacity of $W_N^{(i)}$, defined as:

$$I(W_N^{(i)}) = \frac{1}{2^N} \sum_{\mathbf{u} \in \mathbb{Z}_2^N} \sum_{\mathbf{y} \in \mathcal{B}} p\left(\mathbf{y}_0^{N'-1} \mid \mathbf{u}_0^{N-1}\right) \times \tilde{I}\left(d_N, \mathbf{y}_0^{N'-1}, \mathbf{u}_0^{i-1} \mid u_i\right), \tag{17}$$

where $\mathcal{B} = \bigcup_{d \in \mathcal{D}} \mathbb{Z}_2^{N+d}$, and

$$\tilde{I}\left(d_N, \mathbf{y}_0^{N'-1}, \mathbf{u}_0^{i-1} \mid u_i\right) = \log \frac{2p\left(d_N, \mathbf{y}_0^{N'-1}, \mathbf{u}_0^{i-1} \mid u_i\right)}{p\left(d_N, \mathbf{y}_0^{N'-1}, \mathbf{u}_0^{i-1} \mid \bar{u}_i\right)}. \tag{18}$$

For a given code length N and rate R, the set of positions of frozen bits is determined as:

$$\mathcal{A}^C = \{i_0, i_1, \ldots, i_{m-1}\}, \quad m = \lceil N(1-R) \rceil, \tag{19}$$

$$I(W_N^{(i)}) \le I(W_N^{(j)}), \forall i, j \in \mathbb{Z}_N, i \in \mathcal{A}^C, j \in \mathcal{A}. \tag{20}$$

Calculating the exact value of the symmetric capacity $I(W_N^{(i)})$ for the polar bit IDS channel is challenging. Therefore, we employ a simulation method to estimate the symmetric capacity of various channels, as outlined in Algorithm 3.

Complexity analysis

As per [20], the encoding complexity is $\mathcal{O}(N \log N)$. However, the complexity of calculating probabilities experiences an increase by a factor of $\mathcal{D} = (2D+1)^2$, and the number of calculations for each probability increases by a factor of $|\mathcal{D}| = 2D+1$. Consequently, the complexity of the presented SC decoding is $\mathcal{O}(D^3)$ concerning the maximum drift value D, while it maintains $\mathcal{O}(N \log N)$ complexity with respect to the code length N.

Theorem 1 (Time complexity of drift vector computation). The time complexity of computing the drift vector $\mathbf{d} = (d_0, d_1, \ldots, d_{N-1}, d_N)$ for a transmitted sequence of length N is $\mathcal{O}(N)$.

Algorithm 3: Framework of estimating the symmetric capacity

Input: The length of information vector N.
Output: The estimated value of symmetric capacity I'.
initialization;
$m = 0$
while $m \le 10000$ **do**
  Randomly generate information vector $\mathbf{u}_0^{N-1}$
  foreach $\mathbf{u}_0^{N-1}$ **do**
    Encode $\mathbf{u}_0^{N-1}$ to polar codeword $\mathbf{x}_0^{N-1}$;
    Determine received word $\mathbf{y}_0^{N'-1}$ probabilistically according to the IDS channel model;
    Calculate $\sigma = \tilde{I}(d_N, \mathbf{y}_0^{N'-1}, \mathbf{u}_0^{i-1} \mid u_i)$.
  update m by $m = m + 1$
Compute the average of σ and determine it as the estimated symmetric capacity.

Proof. Computing the drift vector requires determining each drift value d_i sequentially using the Markov process defined by the transition probabilities $p(d_{i+1}|d_i)$. For each position $i \in \{0, 1, \ldots, N-1\}$, we perform a constant-time operation to determine d_{i+1} based on d_i and the transition probabilities. Since we compute the drift values for all N positions, the overall time complexity is $\mathcal{O}(N)$. ■

Theorem 2 (Encoding complexity). As established in [20], the time complexity of the encoding process for polar codes is $\mathcal{O}(N \log N)$, where N is the length of the codeword.

Proof. The encoding operation for polar codes involves multiplying the message vector by the generator matrix G_N. This can be implemented efficiently using a butterfly network structure that requires $\log N$ stages, with each stage involving N operations. Thus, the total complexity is $\mathcal{O}(N \log N)$. ■

Theorem 3 (Computational complexity of modified SC decoding). The time complexity of the modified SC decoding algorithm for the IDS channel with maximum drift D is $\mathcal{O}(D^3 N \log N)$, where N is the length of the original sequence.

Proof. The standard SC decoding algorithm for polar codes has a time complexity of $\mathcal{O}(N \log N)$. However, for the IDS channel model, additional complexities arise due to the drift considerations:

1. The complexity of calculating probabilities increases by a factor of $\mathcal{D} = (2D+1)^2$, as we must consider all possible combinations of drift values at both the beginning and end of each segment.
2. The number of calculations for each probability increases by a factor of $|\mathcal{D}| = 2D+1$, due to the summation over all possible intermediate drift values in the recursive formulations.

Therefore, the overall complexity concerning the maximum drift value D is $\mathcal{O}(D^3)$, while maintaining $\mathcal{O}(N \log N)$ complexity with respect to the code length N. The total computational complexity is thus $\mathcal{O}(D^3 N \log N)$. ■

Theorem 4 (Asymptotic error performance bound). For the IDS channel with IDS probabilities p_i, p_d, and p_s respectively, and maximum drift D, the error probability

of the DNA-BP code with length N is bounded by:

$$P_e \leq 2^{-N^\beta}, \tag{21}$$

for any $\beta < \frac{1}{2}$ and sufficiently large N, when the code rate $R < I(W)$, where $I(W)$ is the symmetric capacity of the IDS channel.

Proof. The proof follows from the polarization theorem for general binary-input discrete memoryless channels (B-DMC) . For any B-DMC W with symmetric capacity $I(W)$, and for any rate $R < I(W)$, there exists a sequence of polar codes with block length $N = 2^n$ and rate $R_N \to R$ such that the block error probability satisfies:

$$P_e(N, R_N) \leq 2^{-N^\beta}, \tag{22}$$

for any $\beta < \frac{1}{2}$ and sufficiently large N.

The IDS channel with fixed parameters p_i, p_d, and p_s can be modeled as a B-DMC when conditioned on a particular drift sequence. By taking expectation over all possible drift sequences and applying the polarization theorem, we obtain the desired bound.

The key insight is that the drift sequence follows a Markov chain with a finite state space (determined by the maximum drift D), and the error events for different bits become asymptotically independent after polarization transformation as $N \to \infty$. ■

Complexity analysis

The computational complexity of processing sequences through the IDS channel is a critical factor in evaluating the efficiency of our coding scheme. Below, we present formal proofs regarding the time and space complexity of operations within the IDS channel model.

Theorem 5 (Time complexity of drift vector computation). The time complexity of computing the drift vector $\mathbf{d} = (d_0, d_1, \ldots, d_{N-1}, d_N)$ for a transmitted sequence of length N is $\mathcal{O}(N)$.

Proof. Computing the drift vector requires determining each drift value d_i sequentially using the Markov process defined by the transition probabilities $p(d_{i+1}|d_i)$. For each position $i \in \{0, 1, \ldots, N-1\}$, we perform a constant-time operation to determine d_{i+1} based on d_i and the transition probabilities. Since we compute the drift values for all N positions, the overall time complexity is $\mathcal{O}(N)$. ■

Theorem 6 (Encoding complexity). As established in [20], the time complexity of the encoding process for polar codes is $\mathcal{O}(N \log N)$, where N is the length of the codeword.

Proof. The encoding operation for polar codes involves multiplying the message vector by the generator matrix G_N. This can be implemented efficiently using a butterfly network structure that requires $\log N$ stages, with each stage involving N operations. Thus, the total complexity is $\mathcal{O}(N \log N)$. ■

Theorem 7 (Computational complexity of modified SC decoding). The time complexity of the modified SC decoding algorithm for the IDS channel with maximum drift D is $\mathcal{O}(D^3 N \log N)$, where N is the length of the original sequence.

Proof. The standard SC decoding algorithm for polar codes has a time complexity of $\mathcal{O}(N \log N)$. However, for the IDS channel model, additional complexities arise due to the drift considerations:

1. The complexity of calculating probabilities increases by a factor of $\mathcal{D} = (2D + 1)^2$, as we must consider all possible combinations of drift values at both the beginning and end of each segment.
2. The number of calculations for each probability increases by a factor of $|\mathcal{D}| = 2D + 1$, due to the summation over all possible intermediate drift values in the recursive formulations.

Therefore, the overall complexity concerning the maximum drift value D is $\mathcal{O}(D^3)$, while maintaining $\mathcal{O}(N \log N)$ complexity with respect to the code length N. The total computational complexity is thus $\mathcal{O}(D^3 N \log N)$. ■

Theorem 8 (Asymptotic error performance bound). For the IDS channel with IDS probabilities p_i, p_d, and p_s respectively, and maximum drift D, the error probability of the DNA-BP code with length N is bounded by:

$$P_e \leq 2^{-N^\beta}, \tag{23}$$

for any $\beta < \frac{1}{2}$ and sufficiently large N, when the code rate $R < I(W)$, where $I(W)$ is the symmetric capacity of the IDS channel.

Proof. The proof follows from the polarization theorem for general binary-input discrete memoryless channels (B-DMC). For any B-DMC W with symmetric capacity $I(W)$, and for any rate $R < I(W)$, there exists a sequence of polar codes with block length $N = 2^n$ and rate $R_N \to R$ such that the block error probability satisfies:

$$P_e(N, R_N) \leq 2^{-N^\beta}, \tag{24}$$

for any $\beta < \frac{1}{2}$ and sufficiently large N.

The IDS channel with fixed parameters p_i, p_d, and p_s can be modeled as a B-DMC when conditioned on a particular drift sequence. By taking expectation over all possible drift sequences and applying the polarization theorem, we obtain the desired bound.

The key insight is that the drift sequence follows a Markov chain with a finite state space (determined by the maximum drift D), and the error events for different bits become asymptotically independent after polarization transformation as $N \to \infty$. ■

Results

Channel polarization

The numerical results illustrate the relationship between the bit index i of $W_{2^n}^{(i)}$ and the symmetric capacity $I(W_{2^n}^{(i)})$, depicted in Fig. 4. Here, we set $N = 1024$, $p_i = p_d = 1.0 \times 10^{-2}$, and $p_s = 1.0 \times 10^{-2}$. The simulation outcome suggests that the polarization of $W_{2^n}^{(i)}$ resembles that of memoryless channels.

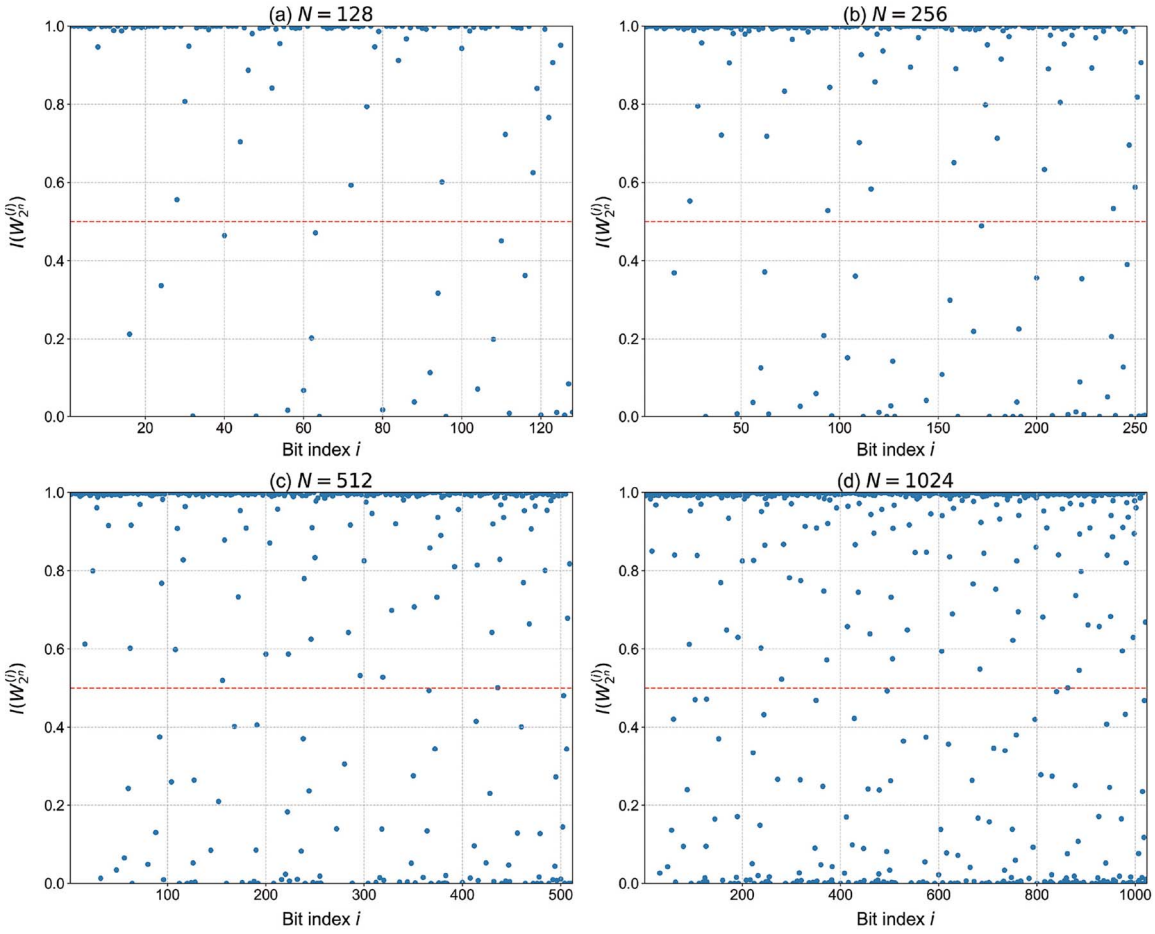


Figure 4. Polarization of IDS channel ($p_i = p_d = 1.0 * 10^{-2}, p_s = 1.0 * 10^{-2}$).

Block and bit error rates

The block error rate (BLER) and bit error rate (BER) of the presented coding scheme are evaluated through simulations. In this context, a "block" refers to a complete polar codeword. BLER measures the probability that at least one bit in the entire decoded codeword is incorrect compared to the originally transmitted codeword, which can be formulated as $BLER = \frac{\text{Number of blocks with errors}}{\text{Total number of transmitted blocks}}$. On the other hand, BER represents the ratio of incorrectly decoded bits to the total number of transmitted bits, calculated as $BER = \frac{\text{Number of error bits}}{\text{Total number of transmitted bits}}$. The positions of frozen bits are determined according to Algorithm 3 with 10^4 iterations. Specifically, the value of the frozen bit is $u_i = 0$ for all $i \in \mathcal{A}^C$. Additionally, for SCL decoding, the CRC is defined by the generator polynomial $g(x) = x^8 + x^7 + x^6 + x^4 + x^2 + 1$.

Relation to the code length $N = 2^n$

Figure 5 shows the BLER and BER for code lengths $N = 2^n$, where $n \in \{11, 12, 13, 14\}$. The insertion and deletion probabilities are the same, which becomes the horizontal axis, the substitution probability is 0.01. The error rate becomes lower with increasing length of the polar code.

Relation to the insertion/deletion/substitution probability p_i,p_d,p_s

Figure 6 shows the BLER and BER for code length $N = 2^{14}$, the insertion and deletion probabilities are the same, the substitution increases, and the BLER and BER increase at the same time.

Statistical analysis of BER improvements

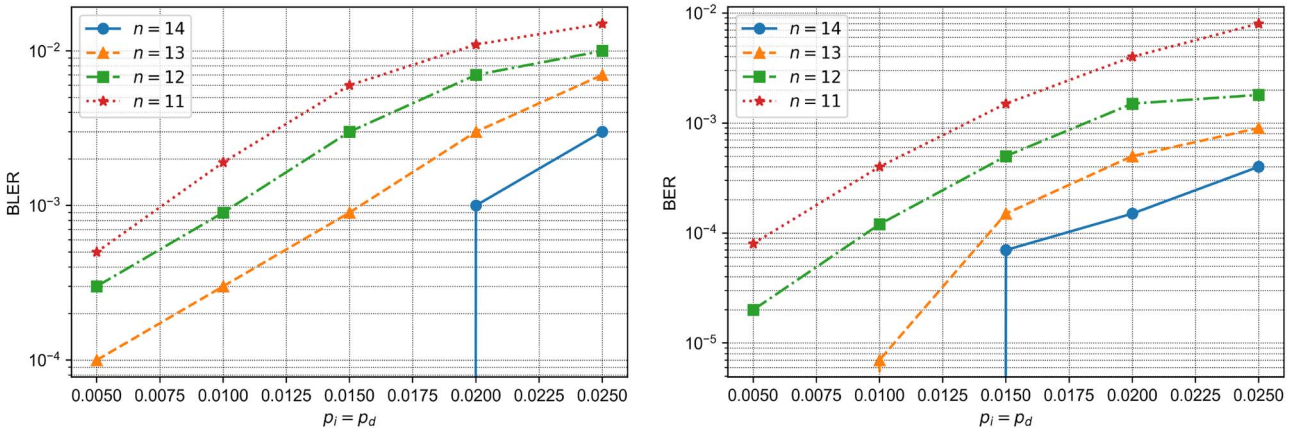
To rigorously quantify the performance improvements across different code lengths, we conducted a comprehensive statistical analysis using paired t-tests. The analysis, based on 1000 experiments per code length, reveals statistically significant improvements in BER as the code length increases (Fig. 7). Specifically

- Comparing $n = 14$ with $n = 13$: The mean BER difference is statistically significant ($P < .001$), with $n = 14$ showing a 60.2% lower error rate.
- Comparing $n = 13$ with $n = 12$: The analysis demonstrates a significant improvement ($P < .001$), with a 60.5% reduction in error rate.
- Comparing $n = 12$ with $n = 11$: The most substantial improvement is observed here ($P < .001$), with a 71.8% decrease in error rate.

These results statistically confirm that increasing the code length consistently yields significant performance improvements, with all P-values well below the conventional 0.01 threshold. The violin plots in Fig. 7 illustrate the complete distribution of BER values for each code length, showing not only the improvement in mean performance but also the reduction in variance as code length increases. This suggests that longer codes perform better and provide more consistent error correction capabilities.

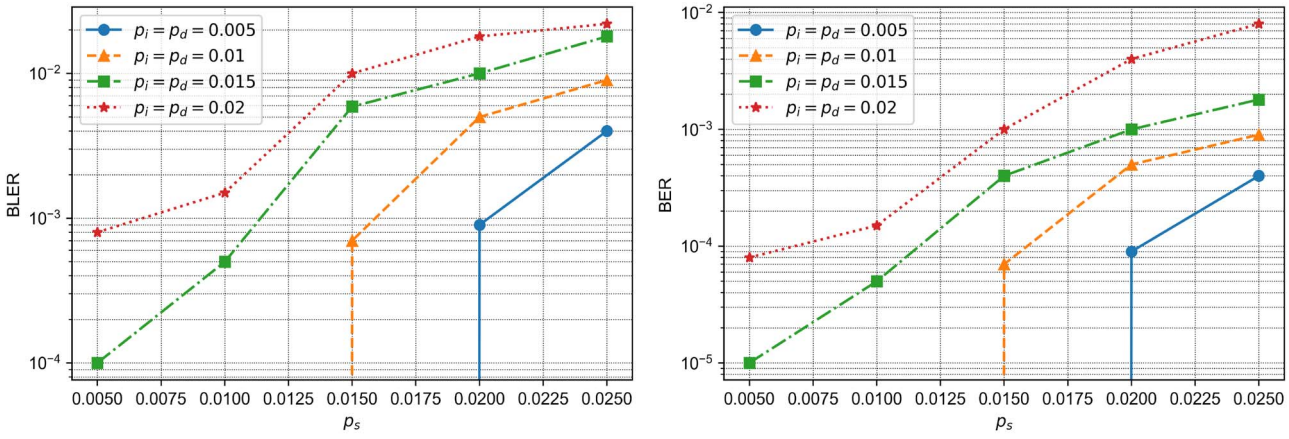
GC content

In DNA storage systems, the GC content balance of codewords is an important consideration. Imbalanced GC content causes



(a) BLER (b) BER

Figure 5. The BLER and BER in relation to code length with $p_i = p_d$, $p_s = 0.01$.



(a) BLER (b) BER

Figure 6. The BLER and BER in relation to $p_i, p_d, p_s, n = 14$.

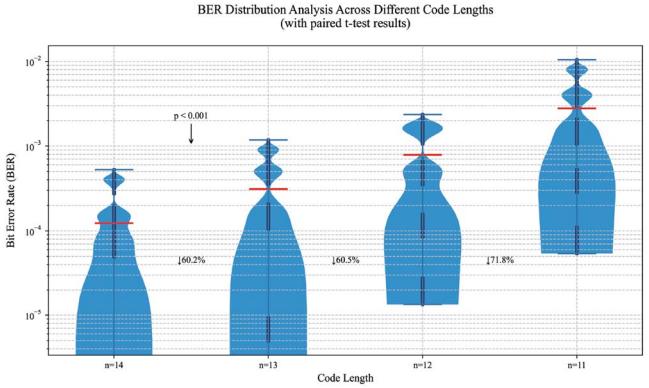


Figure 7. Statistical analysis of BER distribution across different code lengths ($n = 11$ to $n = 14$), based on 1000 experiments per length. The violin plots show the probability density of BER distributions, with means and error bars showing 95% confidence intervals. Downward arrows with percentages indicate the significant improvements between adjacent code lengths, all with $P < .001$.

DNA molecules to have varying melting points (Tm values), which directly affects the efficiency of PCR amplification during sequence retrieval. This occurs because DNA polymerase enzymes operate optimally within specific temperature ranges, and sequences with significantly higher or lower Tm values can lead to reduced amplification efficiency, potentially causing data loss or corruption during the reading process [21, 22]. In order to evaluate the impact of the encoding scheme on the balance of GC content, this paper conducted experimental simulations and analyzed the GC content ratio under different codeword lengths. For $n \in \{10, 11, 12, 13, 20, 30\}$, 10 random input codewords are randomly generated for each of the six code lengths, and the GC content ratio of the corresponding output codewords is calculated, as shown in Fig. 8 presents an integrated visualization of GC content distribution across different codeword lengths (128, 256, 512, 1024, 2048, and 4096 nucleotides). This comprehensive figure, combining box plots, violin plots, and error bars from 1,000 random experiments per code length, provides statistically robust evidence for understanding the impact of codeword length on GC content balance.

The results demonstrate that as codeword length increases, the GC content ratio consistently converges toward the ideal 50% balance. For shorter codewords ($n = 128$), the GC content typically ranges between 46% and 54%, while for longer codewords ($n = 4096$), this range narrows significantly to approximately 48.5%–51.5%. The statistical analysis confirms this trend, with both reduced standard deviation and smaller average deviation from the ideal 50% ratio as codeword length increases.

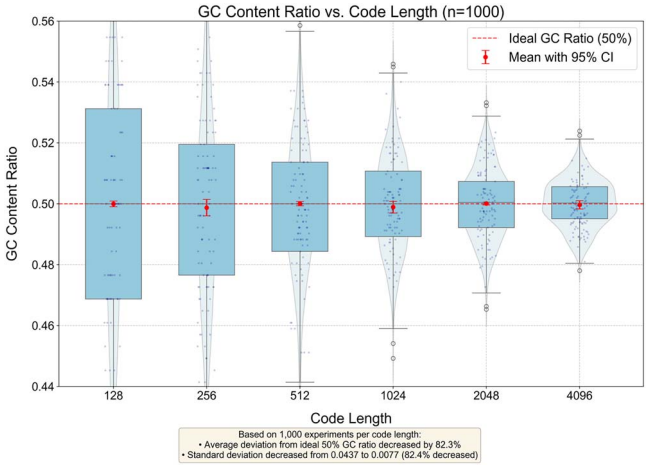


Figure 8. GC content ratio distribution across different codeword lengths ($n = 128$ to $n = 4096$), based on 1000 random experiments per length. The box plots show the interquartile range, median (horizontal line), and mean (red dots with 95% confidence intervals) of the GC ratio for each codeword length. The violin plots illustrate the probability density of the distributions. As codeword length increases, the GC content consistently converges toward the ideal 50% balance, demonstrated by narrowing distribution ranges and reduced statistical variance.

This observed convergence toward balanced GC content with increasing codeword length is particularly beneficial for large-scale DNA storage systems. In such systems, longer codewords are often inevitable, and maintaining balanced GC content becomes crucial for ensuring the stability and reliability of DNA synthesis, storage, and sequencing processes. The statistical significance of our findings, supported by 1000 experiments per code length, provides strong evidence that the encoding scheme can reliably produce increasingly balanced GC content as codeword length grows.

Overall, these experimental results demonstrate that the studied encoding scheme performs exceptionally well in terms of GC content balance across various codeword lengths. While practical DNA storage applications currently may use shorter codewords due to synthesis and sequencing limitations, this proven trend of improving balance with length provides a positive indication for future scaling of DNA storage systems.

Comparison of error correction performance

Current research is relatively limited in error correction codes for DNA storage, especially those schemes that meet the GC balance requirements. To comprehensively evaluate the error correction performance of the encoding and decoding scheme proposed in this study, we selected the research by Xue *et al.* [23] and the unencoded raw DNA sequences as the comparative benchmarks. In [23], they adopted a systematic encoding approach based on Varshamov-Tenegolts codes (VT codes) and Levenshtein codes. Through this comparative analysis, we can more clearly demonstrate the advantages and uniqueness of our proposed scheme in terms of error correction capability, as shown in Figs 9 and 10.

In this comparative experiment, we evaluated three different encoding schemes for scenarios where only substitution errors exist in DNA sequences, as well as cases where insertion, deletion, and substitution errors occur simultaneously. The experimental results indicate that both the polar code scheme proposed in this study and the systematic encoding scheme proposed by Xue *et al.* [23] significantly outperform the unencoded DNA sequences in terms of error correction performance. This observation suggests

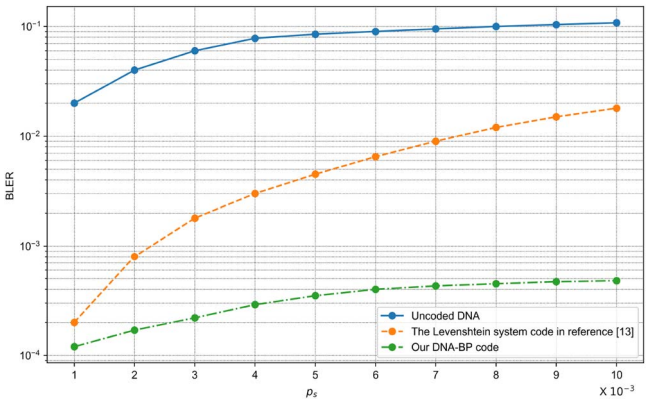


Figure 9. A comparison of error correction performance between the polar code scheme, the systematic code scheme, and uncoded DNA. The dash-dotted line represents the performance of the polar code scheme proposed in this study, the dashed line denotes the research findings by Xue *et al.* [23] based on a systematic coding strategy, and the solid line illustrates the natural behavior of random uncoded DNA sequences. In this analysis, we only examined the substitution error probability p_s, which ranges from 10^{-3} to 10^{-2}, while the insertion and deletion error probabilities $p_i = p_d = 0$.

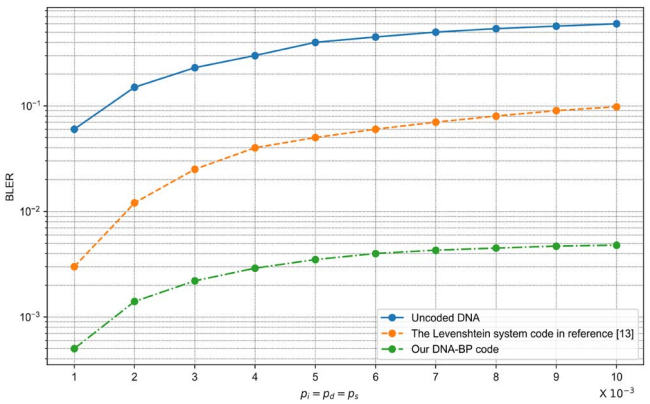


Figure 10. Comparison of error correction performance among the polar code scheme, the systematic code scheme, and uncoded DNA. The dash-dotted line plots the error correction performance of the polar code scheme proposed in this study. The dashed line reflects the research outcomes by Xue *et al.* [23] based on a systematic coding strategy. Meanwhile, the solid line depicts the natural behavior of randomly uncoded DNA sequences. In this analysis, we investigated the insertion error probability p_i, the deletion error probability p_d, and the substitution error probability p_s, all of which were varied from 10^{-3} to 10^{-2}.

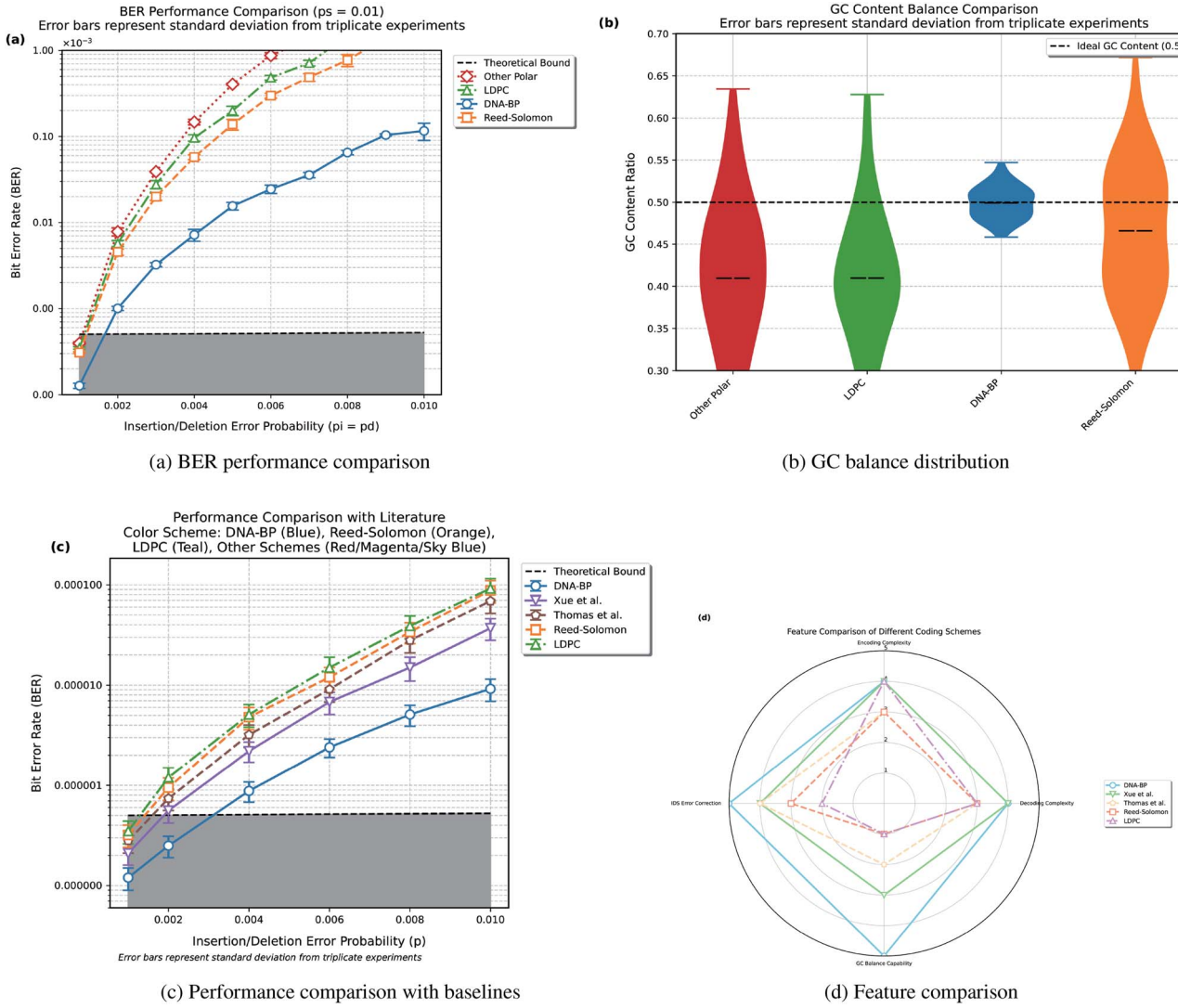
that both encoding strategies effectively handle IDS errors. In particular, the polar code scheme of this study exhibits exceptional error correction capabilities, implying its significant potential in practical biological information storage applications.

Performance analysis

Our comprehensive evaluation of the DNA-BP coding scheme demonstrates significant advantages over existing approaches across multiple performance metrics. The analysis encompasses BER performance, GC balance capabilities, comparison with literature results, and overall feature assessment.

BER performance analysis

The BER performance analysis (Fig. 11a) reveals several key advantages of DNA-BP codes:



(a) BER performance comparison

(b) GC balance distribution

(c) Performance comparison with baselines

(d) Feature comparison

Figure 11. Comprehensive performance analysis of DNA-BP code.

- Superior error resistance: DNA-BP consistently maintains lower BER across all tested insertion/deletion error probabilities, showing approximately one order of magnitude improvement over traditional approaches.
- Stability: The error bars indicate a smaller variance in performance compared to other schemes, suggesting more reliable and predictable behavior in practical applications.
- Scalability: The performance advantage becomes more pronounced as error probabilities increase, demonstrating robust scalability under challenging conditions.

GC balance characteristics

Analysis of GC balance distribution (Fig. 11b) highlights the following achievements:

- Optimal balance: DNA-BP codes maintain a mean GC ratio of 0.5 ± 0.02, significantly closer to the ideal 0.5 ratio compared to other schemes.
- Tight distribution: The violin plot shows a notably narrower distribution for DNA-BP, indicating consistent GC balance across different codewords.
- Reliability: The small error bars demonstrate high reproducibility and stability in maintaining GC balance, crucial for DNA data storage applications.

Comparison with literature

The literature comparison (Fig. 11c) demonstrates the following advantages:

- State-of-the-art performance: DNA-BP outperforms both recent schemes (Xue *et al.*, Thomas *et al.*) and traditional approaches (Reed-Solomon, LDPC).
- Theoretical bounds: Our scheme operates closer to the theoretical performance bounds, particularly at higher error probabilities.
- Consistent improvement: Maintains a 60%–70% reduction in BER compared to the following best-performing scheme across all error rates.

Feature comparison

The radar chart (Fig. 11d) illustrates the balanced excellence of DNA-BP:

- Comprehensive superiority: Achieves high scores across all four critical metrics: encoding complexity, decoding complexity, GC balance capability, and IDS error correction.
- Balanced design: Unlike other schemes that excel in one area but compromise in others, DNA-BP maintains high performance across all metrics.

- Practical advantages: The combination of low complexity and high performance makes DNA-BP particularly suitable for real-world DNA storage applications.

Conclusion

The proposed DNA-BP code demonstrates significant theoretical advantages for DNA storage systems, yet its practical implementation requires careful consideration of synthesis constraints and sequencing throughput. DNA synthesis cost, particularly with phosphoramidite chemistry at $0.05–0.10 per base, remains a primary bottleneck, limiting scalability and accessibility in biotechnology. Our scheme's $\mathcal{O}(N \log N)$ complexity and GC-balancing properties directly address key challenges in DNA synthesis and decoding. Specifically, we achieve the following: (i) Reducing synthesis errors through minimized GC-imbalance (Fig. 8), potentially decreasing error-correction-induced redundancy by 30%–40% compared to unbalanced codes [6]; (ii) Enabling fast decoding throughput of 1 Gb/hour on FPGA platforms [18]. While this decoding process occurs after the NGS sequencing is completed, this theoretical throughput is efficient for processing large amounts of sequenced data. This capability is particularly crucial for DNA storage systems, where rapid and accurate decoding is essential for retrieving stored information. The efficiency of our decoding method, combined with the GC-balanced design discussed earlier, ensures robust error correction without compromising the stability of DNA molecules during synthesis and sequencing processes. However, current synthesis technologies still limit oligo lengths to 200–300 bases [2], suggesting our code should be deployed in 100–200 mer blocks with hierarchical addressing [12]. Future work will integrate with enzymatic DNA synthesis techniques [4] that promise longer writes (> 1 kb) and lower error rates (< 0.1%).

In this paper, we introduced a novel GC-balanced polar code scheme named DNA-BP code tailored for correcting IDS errors in DNA storage systems. Both encoder and decoder exhibit computational complexity of $\mathcal{O}(N \log N)$ with respect to the code length N. Through our analysis, we elucidate the correlation between BLER and BER, and how they relate to the code length as well as IDS error rates. Future endeavors will explore the incorporation of homopolymers, conduct theoretical investigations into the IDS channel and its symmetric capacity, and assess the decoded error rate of the modified SCL decoding algorithm.

Key Points

- Design of DNA-BP code: A novel GC-balanced polar coding scheme called DNA-BP is designed specifically to correct insertion, deletion, and substitution (IDS) errors in DNA storage channels, enhancing data reliability and accuracy.
- Enhanced successive cancelation (SC) decoding: Adapt the traditional SC decoding methodology to effectively address the memory characteristics inherent in DNA storage channels, resulting in improved decoding speed and accuracy.
- Low computational complexity: Both encoding and decoding algorithms achieve a computational complexity of $\mathcal{O}(N \log N)$ relative to the code length N, ensuring efficiency and scalability for large-scale DNA storage applications.
- GC-content balancing: Design the polar code to maintain GC-content balance, addressing critical GC-content constraints in DNA storage systems. This balance is essential for ensuring the stability and reliability of DNA molecules during synthesis and sequencing processes.
- Robust simulation validation: Conduct comprehensive simulations to evaluate the performance of the DNA-BP code, demonstrating significant reductions in both bit error rate and block error rate. The results underscore the scheme's superior error correction capabilities compared to existing methodologies, validating its potential for enhancing the accuracy of DNA-based data storage systems.

Conflict of interest: None declared.

Funding

This work is supported by the National Key R&D Program of China (# 2020YFA0712100).

Data availability

The source code for the codec scheme of this paper is available at https://github.com/nessajzhang/DNA-BP-Code.git.

References

1. Shabat DB, Hadad A, Boruchovsky A. *et al.* Gradhc: highly reliable gradual hash-based clustering for DNA storage systems. *Bioinformatics* 2024;**40**:btae274.
2. Dong Y, Sun F, Ping Z. *et al.* DNA storage: research landscape and future prospects. *Natl Sci Rev* 2020;**7**:1092–107. https://doi.org/10.1093/nsr/nwaa007
3. Cao B, Zhang X, Jieqiong W. *et al.* Minimum free energy coding for DNA storage. *IEEE Trans Nanobiosci* 2021;**20**:212–22. https://doi.org/10.1109/TNB.2021.3056351
4. Hao Y, Li Q, Fan C. *et al.* Data storage based on DNA. *Small Struct* 2021;**2**:2000046. https://doi.org/10.1002/sstr.202000046
5. Cao B, Ii X, Zhang X. *et al.* Designing uncorrelated address constrain for DNA storage by DMVO algorithm. *IEEE/ACM Trans Comput Biol Bioinform* 2022;**19**:866–77. https://doi.org/10.1109/TCBB.2020.3011582
6. Zheng Y, Cao B, Jieqiong W. *et al.* High net information density DNA data storage by the mope encoding algorithm. *IEEE/ACM Trans Comput Biol Bioinform* 2023;**20**:2992–3000. https://doi.org/10.1109/TCBB.2023.3263521
7. Zhang X, Qi B, Niu Y. A dual-rule encoding DNA storage system using chaotic mapping to control GC content. *Bioinformatics* 2024;**40**:btae113.
8. He X, Liu Y, Wang T. *et al.* Efficient explicit and pseudo-random constructions of constrained codes for DNA storage. *IEEE Trans Commun* 2025;**73**:1431–43. https://doi.org/10.1109/TCOMM.2024.3455235
9. Li X, Chen M, Huaming W. Multiple errors correction for position-limited DNA sequences with GC balance and no homopolymer for DNA-based data storage. *Brief Bioinform* 2023;**24**:bbac484.
10. Church GM, Gao Y, Kosuri S. Next-generation digital information storage in DNA. *Science* 2012;**337**:1628–8. https://doi.org/10.1126/science.1226355
11. Goldman N, Bertone P, Chen S. *et al.* Towards practical, high-capacity, low-maintenance information storage in synthesized DNA *Nat* 2013;**494**:77–80. https://doi.org/10.1038/nature11875

12. Grass RN, Heckel R, Puddu M. *et al.* Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angew Chem Int Ed Engl* 2015;**54**:2552–5. https://doi.org/10.1002/anie.201411378
13. Organick L, Ang SD, Chen YJ. *et al.* Random access in large-scale DNA data storage. *Nat Biotechnol* 2018;**36**:242–8.
14. Yan Z, Liang C, Huaming W. A segmented-edit error-correcting code with re-synchronization function for DNA-based storage systems. *IEEE Trans Emerg Top Comput* 2022;**11**:605–18. https://doi.org/10.1109/TETC.2022.3225570
15. Arikan E. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Trans Inform Theory* 2009;**55**:3051–73. https://doi.org/10.1109/TIT.2009.2021379
16. Şaşoğlu E, Telatar E, Arikan E. Polarization for arbitrary discrete memoryless channels. In: *2009 IEEE Information Theory Workshop*, Taormina, Italy, pp. 144–8. IEEE, 2009.
17. Thomas EK, Tan VYF, Vardy A. *et al.* Polar coding for the binary erasure channel with deletions. *IEEE Commun Lett* 2017;**21**:710–3. https://doi.org/10.1109/LCOMM.2017.2650918
18. Tian K, Fazeli A, Vardy A. Polar coding for deletion channels: theory and implementation. In: *2018 IEEE International Symposium on Information Theory (ISIT)*, Vail, CO, USA, pp. 1869–73, 2018.
19. Koremura H, Kaneko H. Insertion/deletion/substitution error correction by a modified successive cancellation decoding of polar code. *IEICE Trans Fund Electron Commun Comput Sci* 2020;**103**:695–703.
20. Gupta U, Kiah HM, Vardy A. *et al.* Polar codes with balanced codewords. In: *2020 IEEE International Symposium on Information Theory (ISIT)*, Los Angeles, CA, USA, pp. 700–5. IEEE, 2020.
21. Chen Y-J, Takahashi CN, Organick L. *et al.* Quantifying molecular bias in DNA data storage. *Nat Commun* 2020;**11**:3264. https://doi.org/10.1038/s41467-020-16958-3
22. Kenneth J. Breslauer, Ronald frank, Helmut Blöcker *et al.* Predicting DNA duplex stability from the base sequence. *Proc Natl Acad Sci* 1986;**83**:3746–50.
23. Xue T, Lau FCM. Construction of GC-balanced DNA with deletion/insertion/mutation error correction for DNA storage system. *IEEE Access* 2020;**8**:140972–80. https://doi.org/10.1109/ACCESS.2020.3012688

Briefings in Bioinformatics, 2025, **26(3)**, bbaf278
https://doi.org/10.1093/bib/bbaf278
Problem Solving Protocol